

Priezvisko:		Body
Meno:		

	a	b	c	d	e
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

V otázkach je len jedna možnosť správna. Vyznačte svoju odpoveď krížikom do tabuľky uvedenej nižšie. Hodnotia sa len odpovede vyznačené v tabuľke. V prípade opravy jasne vyznačte ktorú odpoveď vyberáte.

Každá správna odpoveď má hodnotu 1 bod. Nesprávna odpoveď, vyznačenie viac odpovedí alebo nejednoznačné vyznačenie má hodnotu 0 bodov. Postup riešenia sa nehodnotí.

1. Nech  $n$  je pole triedy. Reťazec  $r$  je v jednej z metód tej istej triedy definovaný takto:

```
Integer r[] = new Integer[n];
```

Táto definícia je v Jave

- (a) nekorektná
- (b) korektná jedine ak je pole  $n$  statické
- (c) korektná
- (d) korektná jedine ak je pole  $n$  finálne
- (e) korektná jedine ak je pole  $n$  synchronizované

2. Tok údajov (stream) v Java API sa zatvára

- (a) príkazom `System.close()`
- (b) zavolaním jeho deštruktora
- (c) prečítaním posledného údajov v ňom
- (d) príkazom `IOStream.close()`
- (e) jeho metódou `close()`

3. V programe v Jave sa príkaz

```
package my;
```

- (a) môže vyskytovať len v jednom súbore
- (b) môže vyskytovať vo viacerých súboroch toho istého programu
- (c) môže vyskytovať len v súbore ktorý obsahuje len jednu triedu alebo rozhranie
- (d) môže vyskytovať vo viacerých súboroch toho istého programu len ak sú v tom istom adresári
- (e) nemôže vyskytovať v súboroch

4. Abstraktná trieda v Jave

- (a) nemôže mať metódy
- (b) môže mať len abstraktné metódy
- (c) nemôže mať inicializované polia
- (d) nemusí mať len abstraktné metódy
- (e) nemôže mať prekonávajúce metódy

5. Daný je nasledujúci kód v Jave obsiahnutý v jednej z metód:

```
abstract class A implements I { }  
interface I {  
    void m();  
}  
A[] a = new I[5];
```

Tento kód sa

- (a) preloží a vykoná korektné
- (b) nepreloží, lebo nie je možné definovať triedy a rozhrania v metódach
- (c) preloží, ale vznikne chyba pri vykonávaní posledného riadku
- (d) nepreloží, lebo posledný riadok je zlý
- (e) nepreloží, lebo trieda `A` neimplementuje metódu `m()`

6. Trieda v objektovo-orientovanom programovaní predstavuje

- (a) objekt
- (b) inštanciu
- (c) rozhranie objektu
- (d) interakciu objektov
- (e) typ objektu

7. Prístup `private` je vhodné použiť pri takých prvkoch triedy ku ktorým chceme pristupovať len

- (a) v danej triede
- (b) v odvodených triedach
- (c) v odvodených triedach a v triedach toho istého balíka
- (d) v triedach toho istého balíka
- (e) v odvodených triedach toho istého balíka

8. Dá sa v Jave urobiť inštancia rozhrania?

- (a) áno
- (b) áno, ale len ak neobsahuje metódy
- (c) nie
- (d) áno, ale nebudú sa dať zavolať metódy
- (e) áno, ale len ak obsahuje výlučne statické prvky

9. Daný je nasledujúci program v Jave:

```
class C {  
    C x;  
    void m() {  
        x = new SubC();  
        **1**.f();  
    }  
    public static void main(String[] args) {  
        **2**.m();  
    }  
}
```

```

}

class SubC **3** {
    void f() {
        System.out.println("f");
    }
}

```

Ktoré fragmenty kódu treba v tomto programe doplniť, aby sa pri jeho vykonaní vypísalo f?

- (a) **\*\*1\*\***: x     **\*\*2\*\***: new C()  
      **\*\*3\*\***: extends C
- (b) **\*\*1\*\***: x     **\*\*2\*\***: new C()  
      **\*\*3\*\***: implements C
- (c) **\*\*1\*\***: ((SubC)x)     **\*\*2\*\***: new C()  
      **\*\*3\*\***: nič
- (d) **\*\*1\*\***: ((SubC)x)     **\*\*2\*\***: new C()  
      **\*\*3\*\***: extends C
- (e) **\*\*1\*\***: SubC     **\*\*2\*\***: C  
      **\*\*3\*\***: extends C

10. Dá sa v Jave niť vytvoriť implementáciou rozhrania Runnable?

- (a) áno, ale na spustenie nite sa musí použiť trieda Thread
- (b) nie
- (c) áno, ale len atomická
- (d) áno, ale len ako daemon
- (e) áno, ale len synchronizované

11. Daný je nasledujúci kód:

```

class E1 extends Exception {}
class E2 extends E1 {}

```

```

class A {
    void m() throws E1 {
        ...
    }
}

```

```

class B extends A {
    void m() throws E2 {
        ...
    }
}

```

Metóda m() triedy B

- (a) musí deklarovať že vyhadzuje výnimku typu E1
- (b) je korektná
- (c) nesmie deklarovať že vyhadzuje žiadnu výnimku
- (d) musí ošetrovať výnimku typu E1
- (e) musí vyhadzovať výnimku typu E1

12. Liskovej princíp substitúcie

- (a) predstavuje kritérium pre použitie dedenia
- (b) hovorí o zapuzdrení
- (c) stanovuje kritéria pre použitie abstraktných tried
- (d) predstavuje kritérium pre použitie agregácie
- (e) odčleňuje správanie objektu od jeho implementácie

13. Daný je kód v Jave na obr. 1. Čo sa vypíše po vykonaní týchto príkazov:

```

I o = new B();
o.m();
((A)o).m();
((I)o).m();

```

- (a) bab
- (b) bbb
- (c) aaa
- (d) baa
- (e) bba

---

```

interface I {
    void m();
}

```

```

class A implements I {
    public void m() { System.out.print("a"); }
}

```

```

class B extends A {
    public void m() { System.out.print("b"); }
}

```

---

Obrázok 1: Kód pre otázky 13, 15 a 14.

---

14. K triedam z obr. 1 je daný nasledujúci kód:

```

List<I> z = new ArrayList<A>();
z.add(new B());

```

Tento kód sa

- (a) nepreloží, lebo do zoznamu list sa dajú vkladať len objekty typu A
- (b) nepreloží, lebo typ referencie z nezodpovedá typu priradeného objektu
- (c) preloží, ale padne počas vykonávania s výnimkou ClassCastException
- (d) preloží a vykoná korektné
- (e) nepreloží, lebo trieda ArrayList nie je generická

15. Daný je kód v Jave na obr. 1. Dá sa z metódy m() triedy B zavolať rovnomená metóda triedy A?

- (a) nie
- (b) áno, príkazom m();
- (c) áno, príkazom super.m();
- (d) áno, príkazom A.m();
- (e) áno, príkazom this.m();

- 1 c
- 2 e
- 3 b
- 4 d
- 5 d
- 6 e
- 7 a
- 8 c
- 9 d
- 10 a
- 11 b
- 12 a
- 13 b
- 14 b
- 15 c